

The Horde3D Scene Editor v.0.7.0

Volker Wiendl

April 24, 2008

Contents

1	Horde3D Scene Editor	3
1.1	Installation	3
1.1.1	Windows	3
1.1.2	Linux	4
1.2	Mac	4
2	The first start	4
2.1	Open a Scene	5
2.2	Navigation	5
2.3	Customizing the Editor	7
2.4	Manipulating objects	7
2.4.1	Reference Nodes	7
2.4.2	Camera Nodes	8
3	A new scene	8
3.1	Before you start	8
3.2	Creating a scene	9
3.3	Filling the scene	10
4	Lua Script Interface	11
5	PlugIns	12
6	To be continued...	12

1 Horde3D Scene Editor

The Horde3D graphics engine <http://www.horde3d.org>¹ is a SDK for rendering three dimensional data. It was developed by Nicolas Schulz and is supported by the University of Augsburg. The interface to the graphics data is defined by a Horde3D specific file format. Normally you will create a 3D model in your favorite modeling tool like 3D Max or Blender and export the data to a Collada file. This file can then be converted by the Horde3D Collada converter to the internal Horde3D format.

But what if you want to compose a scene where several single models should be integrated. The first possibility is to use your modeling tool, compose everything in it and export the whole scenery to create one big Horde3D file. But doing it this way suffers still from the lack of possibility to illuminate your scene with Horde3D lights and visualize it with your custom shaders. That's why I created the second possibility: the Horde3D Scene Editor!

It will create scene graph files readable by Horde3D, let you integrate your custom attachment nodes by using plugins, offers fast prototyping by providing a Lua interface to all Horde3D functions and also let you instantly preview changes to shader code, materials, effects, textures and pipelines.

1.1 Installation

For the Horde3D Scene Editor a graphical installer is provided for the Microsoft Windows platform. You can download the latest release at the Horde3D enhancements website². Currently the Editor supports two different platforms: Windows and Linux! It may be used on the Macintosh platform as well, but since I don't have access to a Mac I haven't tested this yet.

1.1.1 Windows

After downloading and extracting the compressed archive you simply have to run the Setup.exe. It will guide you through the rest of the installation procedure. After installation has been done, a Horde3D Scene Editor link has been created in your start menu and the file extension *.scn* has been associated with the editor. The installation directory will contain an executable as well as the source code and a Visual Studio 2005 solution.

¹i

²<http://mm-werkstatt.informatik.uni-augsburg.de/horde3d>

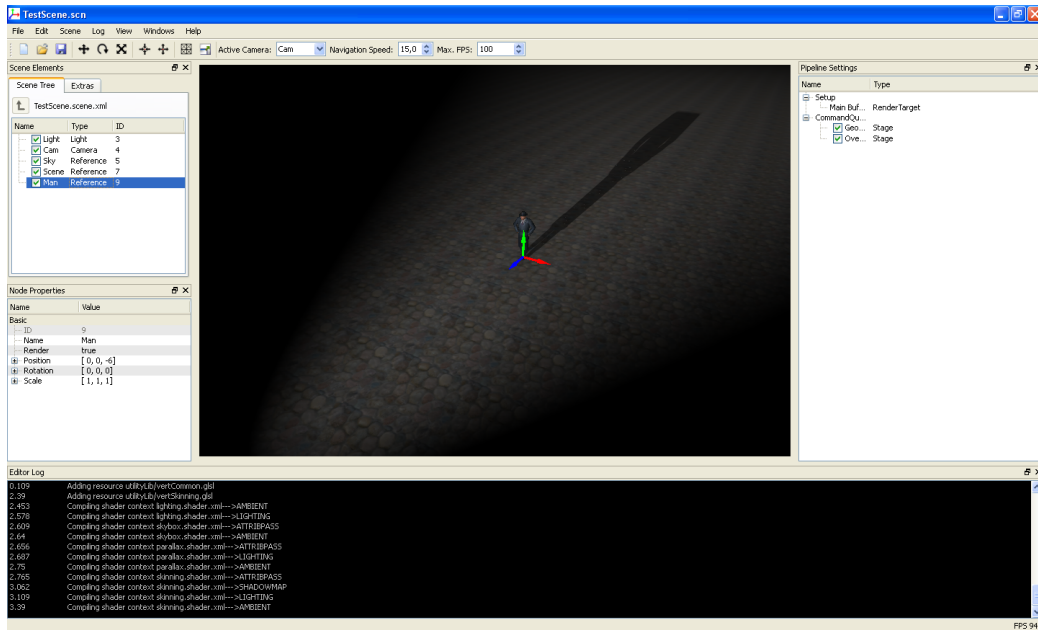


Figure 1: The installation package contains two small test scenes. One of them integrates a camera, a skybox, a man and a floor as well as a light source.

1.1.2 Linux

For the installation under a Linux environment you have to install at least Qt 4.3.3 to be able to compile the editor. After downloading and extracting the compressed archive you have to run *qmake*, which will create the make files for your compiler. By typing *make* afterwards the editor will be created in the *bin* subfolder of the directory where you placed the extracted files.

1.2 Mac

Although it may be possible to compile the linux package under Mac OS X, I can't provide support until someone buys me a macintosh. You can find some further information in the Horde3D forums.

2 The first start

After you called the executable, the application starts and will present you the Horde3D Scene Editor with an empty window. During the installation

two test scenes have been installed to the *bin\TestScene* and *bin\Particle* folder of the Horde3D Scene Editor installation directory. They contain a small scenery to provide you with out-of-the-box examples.

2.1 Open a Scene

To open it click the file menu and select *Open Scene* or press CTRL + O as a shortcut.

For the first test, browse to the TestScene folder and choose the *TestScene.scn* file. After you confirmed your selection with the open button the editor will load the scene and present it to you as you can see it in fig. 1.

2.2 Navigation

There are two possibilities to navigate through the scene. The first one is similar to most of the today's first person shooters. With the key combination *W, A, S, D* and by clicking with the left mouse button in the viewport widget you can move the camera through the scene (note that you have to hold the mouse button pressed to rotate the camera). If you would like to use another mouse button you can reconfigure this in the *Settings* dialog accessible via the *File* menu.

The second possibility is to use the mouse wheel to navigate forward and backward and strafing left, right, up, down by holding the CTRL key and left mouse button pressed and moving the mouse in the direction to strafe. You can change the amount of movement between two frames by changing the *Navigation Speed* entry within the tool bar. Additionally if you keep the *Shift* button pressed the navigation speed will be increased as long as you hold it.

A third possibility was introduced with version 0.7.0 if you hold both the mouse navigation button (normally the left one) and the select button (by default the right one) pressed, you can also strafe along the X- and Y-axis of the camera.

The transformation changes to the camera are done only within Horde3D and not saved to the XML file. To change the transformation of a camera permanently you have to select it first in the scene tree and after you moved it to the place you want, you have to press the *Move Node To Camera Position* button or activate the corresponding menu entry in the *Edit* menu (Shortcut: CTRL+ALT+C). Now if you press the save button, the transformation of the camera will be stored in your scene graph file.

To get a better impression of the scene you can switch the view to fullscreen mode by clicking on the fullscreen button in the toolbar or activate the cor-

responding entry in the *View* menu. To leave the fullscreen mode you have to press the *Escape* key on your keyboard.

Finally the *Wireframe Mode* button toggles between a wireframe rendering mode and the normal rendering. It is especially useful if you have problems with the light and are not sure if a geometry was inserted correctly or not. For the alignment of objects a parallel projection mode for the cameras has been integrated. You can access it using the tool buttons in the camera navigation bar (see 2). But be aware that the parallel projection might cause confusion since the distance of the camera to the objects is only observable via the near clipping plane. So if you move the camera position using the *W* or *S* key, you might get rid of you models because they are clipped by the viewing frustum. To zoom when using the parallel projection use the mouse wheel.

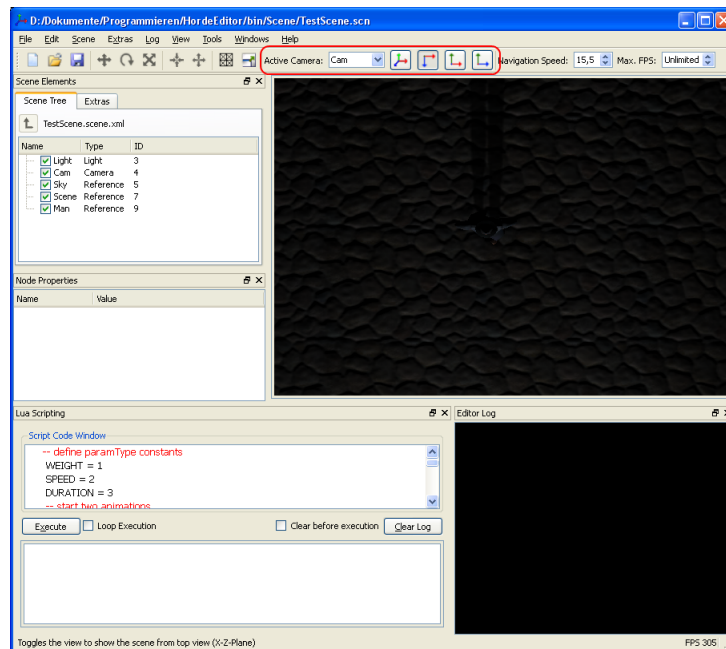


Figure 2: The Camera toolbar allows quick access to all cameras in the scene. Additionally top, front, and right view buttons allow the use of parallel projection.

2.3 Customizing the Editor

For a better workflow you can adjust the dock widgets of the editor to your personal preferences. To do this just click and drag the titlebar of the dock

widgets and place them where you want them to be. By default the most important dock widgets are visible to you. If you want reduce the required screen space there is an option in the *Windows* menu to toggle the dock widget visibility depending on the selected scene graph node. By doing this the attachment widget or material widget are hidden if no plugin (see 5) is loaded or no node with a material attribute has been selected.

You can also alter the way mouse buttons are used for camera navigation and object selection in the settings dialog you can access via the *Settings* entry in the file menu.

2.4 Manipulating objects

After loading the scene file, the elements of the scene are listed in the *Scene Tree*. By selecting one of them, the corresponding transformation - stored within the XML file - is shown in the *Basic* section of the *Node Properties* widget. If you change it, an entry in the *Undo* section of the *Edit* menu is created, allowing you to undo the change. Depending on the type of node you selected, the *Node Properties* widget shows you additional configuration settings. For example if you select a light node you can change the light's color, field of view or many other settings. Nodes referencing a material file cause the *Material Settings* widget to display the content. But take note, that the changes you made to the materials can not be undone by using the undo/redo commands. Since the materials are separated files and not direct part of the scene graph, you have to save them manually (click on the save button) to apply the changes to the scene.

2.4.1 Reference Nodes

A *Reference* node represents a separate scene graph file not directly integrated in the currently shown scene graph tree. To view the content of this file you can double click on the reference element. The scene tree now contains the contents of this file and all changes made on the objects are handled within the scene graph file of this element. If you want to switch back to the parent scene graph file you can click on the *Arrow* at the top of the scene tree or press the backspace key. Take note, that all manipulations are done in the scene graph currently displayed in the tree view. That's why you can only undo actions done within the currently opened scene graph.

If you hold the *CTRL* key pressed, while double clicking on a reference node, the camera will be moved to the selected node.

2.4.2 Camera Nodes

In a scene you can define multiple cameras. To activate one of them you have to double click on the scene tree item in the tree view or select it in the *Active Camera* combobox of the navigation bar. The active camera can also be used to set transformations of other nodes. If you navigate the camera to a special point you can adjust the transformation of the currently selected node by using the *Move Node To Camera Position* button or by pressing the shortcut: CTRL+ALT+C

Orthogonal to this you can set the camera transformation to the transformation of the currently selected node by using the *Move Camera To Node* button.

3 A new scene

Before you create a new scene you may want to configure an editor repository.

3.1 Before you start

This will be done in the *Settings* Dialog (see fig. 3) you can find under the *Settings* menu entry in the *File* menu. There are six different repository paths

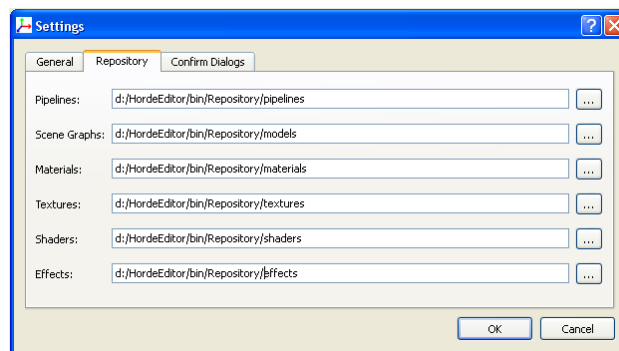


Figure 3: Next to the several other settings you can define a repository for your Horde3D model data in the settings dialog.

to configure. The first is the path to Horde3D pipelines. These files configure the way Horde3D renders the scene. The next path handles the scene-graph (Extension .scene.xml) and Horde3D geometry files (Extension .geo). The third is the path to the materials directory that contains the subfolders and material files (Extension .material.xml) for the materials of a converted 3d model. The texture directory defines the path where the textures are stored

while the shaders directory points to the shader (Extension `.shader.xml`) and code files (Extension `.glsl`) used by the models stored in the repository. Last but not least for effect files (`.effect.xml`) a separate effect directory can be configured. By default the repository is located in the `bin\Repository` folder of the Horde3D scene editor and contains the Horde3D samples' models. Next to the repository configuration you can also define a shader editor in the general tab of the settings dialog. This editor is used when clicking on an edit button in the material editor. Other settings are the path to the plugin directory used to search for attachment plugins (see section 5) and the number of undo steps that will be available. Note that if you choose a great number of undo steps or even the *Unlimited* setting, the Horde3D Scene Editor could consume a lot of memory if you make many changes to one scene. To prevent this you can manually clean the undo stack just by reopen the current scene (every time a scene is closed the undo stack will be cleared).

3.2 Creating a scene

To create a new scene select the *New Scene* entry in the file menu or press the CTRL + N shortcut. A wizard will appear letting you choose a scene file name and a directory where the newly created scene will be stored. If the directory does not exist you will be prompted if you want to create it.

After choosing the scene directory additional directories for the different scene resources have to be selected on the next page. As before directories not already existing can be created by the editor automatically. Take care that the resource directories should be placed below the parent scene directory and such should be declared relative to this folder.

On the next page you have to configure a default camera used for rendering. While you can keep the default frustum settings, you have to import a render pipeline from the repository configured before. By default some of the Horde3D example pipelines are listed when you select *Import from repository* in the combo box. You can then just press the OK Button or adjust the code in the edit window. Only the content of this window will be used for the new camera. The original file in the repository remains unchanged.

To allow you viewing the geometry placed in your scene and to configure a light source for the preview of models, you will be asked to define a default light on the next page. The settings you have to make here are depending on the pipeline configuration you made on the previous page (see the Horde3D documentation for more details). It is important that you configure this light properly since the *Material*, *Lighting Context* and *Shader Context* parameters are used for the standard light source in the preview window when adding

new models to your scene. If your pipeline settings requires the light to have a material (e.g. when using deferred rendering), you can import one from the repository (see section 3.1). the *Import File Dialog*. The yellow lock next

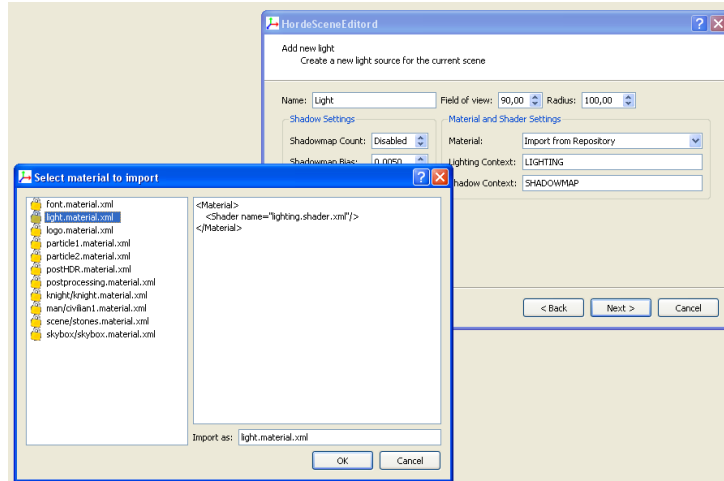


Figure 4: Importing a material from the repository.

to the filename indicates that the material will be imported from the model data repository you specified before. If you have not specified it properly, or do not have a light material, you have to manually copy an appropriate light material and shader file to the specific directories of your scene.

Finally on the last page you can select a plugin for handling Horde3D attachments. If there have been plugins found by the editor, the selection box let you choose one of them to be the attachment controller for this scene. At the moment no public plugin is part of the editor yet.

3.3 Filling the scene

After you created the new scene you will have a black screen because until now there is only a light source and a camera but no content that can be rendered (see fig. 5).

To add real content you can either select one of the add commands in the *Scene* menu or right click in the scene tree window and select one of them there. For example if you choose *Add model* a dialog will pop up, allowing you to select one of the models in the repository (or also any model already existing in your scene) to be imported to the current scene. If you select the *man* from the Horde3D samples to be imported it will be added to your scene. By default all scene objects inserted, in front of the camera with zero rotation.

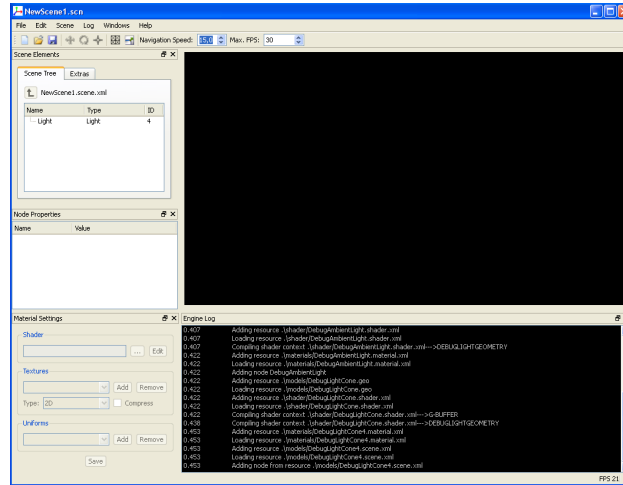


Figure 5: After creating a new scene the screen shows a black window.

If you select the man within the tree, you can move the man by using the mouse if you press the *t*-Key. By default this will transform the selected model parallel to the camera plane. To change the position along a specific direction you can press either *x*, *y* or *z* after you pressed *t* to limit all changes to the corresponding axis (this way of doing transformation changes is inspired by Blender an open source 3D-modeling tool). To confirm the change you have to press the *left mouse button* or the *Return* key. If you want to abort the changes press the *ESC* key or *right mouse button*.

4 Lua Script Interface

To demonstrate the possibilities the Lua Script Interface can offer to you, the second example scene brings a sample script file that will transform some particle emitters in (nearly) real time. To view this scene click the *Open Scene* button and select the Particle.scn file in the *bin\Particle* directory. After opening the file you will first see a black window getting filled with a yellow color. Strange scene you may think, but wait until you opened the lua script. To start it, you first have to make the *Lua Scripting* Window visible by activating the corresponding menu entry in the *Windows* menu. Right click in the *Scrip Code Window* and select *Open* in the context menu. Now choose the *I_love_Horde.lua* script file. Finally check the *Loop Execution* box in the scripting window and wait a few seconds to see the results.

5 PlugIns

Horde3D supports so called attachments. Each node in a scene graph can contain one (and only one) direct attachment child node. These attachments are normally not handled by Horde3D itself but have to be managed by a registered callback function of a library that supports these attachments. To be able to support the functionality of such an attachment library the Horde3D scene editor has an AttachmentPlugIn interface. Each scene file can contain a link to one plugin that will be loaded when opening the scene. The plugin will be searched in the plugin folder you have configured in the *Settings* dialog (see section 2).

6 To be continued...